An analysis of the use of CVEs by IoT malware

Raphaël Khoury[1], Benjamin Vignau[1], Sylvain Hallé[1], Abdelwahab Hamou-Lhadj[2], and Asma Razgallah[1]

[1] Université du Québec à Chicoutimi, Department of Computer Science and Mathematics, Saguenay, Québec, Canada
[2] Concordia University, Department of Electrical and Computer Engineering, Montreal Canada
{raphael.khoury,benjamin.vignau1,asma.razgallah1}@uqac.ca, shalle@acm.org, wahab.hamou-lhadj@concordia.ca

Abstract. In recent years, IoT malware has become a significant threat to the IoT infrastructure, to the point where it even hinders the deployment of this promising technology. A distinctive aspect of this threat is its reliance on vulnerabilities as an infection vector. Many of these vulnerabilities are CVEs (Common Vulnerability Enumeration) selected from the National Vulnerability Database (NVD). In this study, we investigate the use of CVEs by IoT malware, with the ultimate aim of predicting which CVEs are more likely to be targeted by malware developers. Our results show that the CVEs exploited by IoT malware developers are sufficiently distinguished from those CVEs that IoT developers refrain from using to permit effective automated prediction. We detail these differences, develop other observations about the use of vulnerabilities by IoT malware and compile data on this topic that may be useful to security researchers.

Keywords: IoT malware · CVE Internet of Things · malware

1 Introduction

The last ten years have seen an exponential rise in the use of IoT devices: mechanical or digital devices that are connected to a network and can send and receive information without interaction from a user[46]. Unfortunately, this growth has been matched with a corresponding growth in IoT malware, i.e. malware that has been specifically designed to target IoT connected devices. The fact that it is often difficult to update the firmware in such devices makes them a particularly inviting target for malware developers.

Previous research [58] showed that while IoT malware seems to be developed in isolation from malware targeting other device platforms, IoT malware developers borrow freely from each other, re-using code as well as broad features such as infection strategies. A particularly important aspect of the latter is the use of CVEs as an infection vector. CVEs (for Common Vulnerabilities and Exposures) are software vulnerabilities that are documented and given a unique ID for future reference. The National Vulnerability Database (NVD), a database of all CVEs,

is publicly available and maintained by the National Institute of Standards and Technology [43].

In this paper, we investigate the use CVEs as an infection vector by IoT malware. We examine how this infection strategy compares to other strategies employed by IoT malware designers to achieve their nefarious aims, as well as how CVEs are chosen for exploitation. Answers to these questions will allow developers and system managers to better protect their devices against the scourge of IoT malware, for instance by prioritizing CVEs that are more likely to be exploited.

In particular, we attempt to answer the following 3 research questions.

RQ1: What trends are detectable in the use of CVEs by IoT malware? We first attempt to determine how the use of CVEs as an infection vector compares to other infection strategies employed by IoT malware developers. In particular, investigate if the use of CVEs has increased over time; if the certain classes of IoT malware are more likely to employ CVEs, and if vulnerabilities are more likely to be exploited if they are indexed by the NVD.

RQ2: What types of CVEs are targeted by IoT developers? Next, we determine how malware developers choose the CVE entries that are incorporated in their code. We find that CVEs that are chosen by malware developers differs from the broader NVD in several respects, notably w.r.t. impact, complexity, type and date.

RQ3: Can we predict which CVEs will be used by a given malware? Finally, drawing upon the insights gleaned in answering RQ2, we attempt to predict which CVEs are more likely to be exploited in the future using a machine learning process.

The main contribution of this paper is provide answers to the above questions. In addition, we list every CVE exploited by IoT malware during a ten year horizon, and compile other information about the exploitation of vulnerabilities by IoT malware that may be useful to security researchers.

The remainder of this paper is organized as follows: Section 2 presents background about the dataset we utilized. Sections 3, 4, and 5 address RQs 1, 2 and 3 respectively. Section 6 discusses threats to the validity of our conclusions, followed by related works in Section 7. Concluding remarks are given in Section 8.

2 Description of the data

The National Vulnerability Database (NVD) is a freely available database of vulnerabilities, maintained by the National Institute of Standards and Technology, an agency of the United States Department of Commerce [43]. Each entry, called a CVE, captures a single vulnerability, in a any system, and presents it in a standardized format. The NVD contains over 140 000 entries.

The NVD provides standardized information about each entry by way of the Common Vulnerability Scoring System(CVSS). Two versions of the CVSS

are currently in use: Version 2 (V2) introduced in 2007 [32] and Versions 3, introduced in 2015 [18].

Amongst the information provided by the CVSS, the following are particularly relevant to the topic of this study:

Attack Vector Indicates the context by which exploitation can occurs. In the V2 score, this metric ranges over the values 'local', 'adjacent network', and 'network'. The V3 scores adds a fourth value: 'physical'.

Access Complexity This metric captures the presence of conditions beyond the control of the attacker that are nonetheless required to successfully exploit the vulnerability. It ranges over 'low' , 'med' and 'high' in the V2 score, and over 'low' and 'high' in the V3 score.

User Interaction Indicates if user interaction is required in order to successfully exploit the vulnerability. This value ranges over 'true' and 'false' in the V2 score, and over 'none' and 'required' in the V3 score.

Impact score (called impact sub-score in the version 2) A value in the 0-10 range that captures the impact that exploiting the vulnerability may have on the targeted organization.

Exploitability score (called exploitability sub-score in CVSS version 2). It consists in a value in the 0-10 range that captures how vulnerable the system is to attack.

Base score A value in the 0-10 range that captures the severity of the vulnerability. It is derived from the Impact score and Exploitability score using a algorithmic method.

Each CVE possesses a timestamp of the date of publication of the vulnerability and a unique identifier in the format CVE-YYYY-#### where #### is a sequential number. Each CVE also associates a a list of products and vendors that are affected by it. Note that each CVE may be associated with multiple different products, from multiple different vendors.

3 Research Question 1

What trends are detectable in the use of CVEs by IoT malware? We begin by examining what patterns can be discerned in the use of CVEs as an exploitation vector by IoT malware, as opposed to other attack vectors such as credential attacks.

We examined 27 IoT malware, spanning the period 2008 to present. These include every IoT malware that has been studied in the academic literature during this time period. Of these, 13 do not exploit vulnerabilities as part of their infection strategy, opting for other infection mechanisms such as common credentials dictionaries. A single one, Aidra, uses only a single vulnerability not recorded in the CVE database. An additional 6 rely exclusively on vulnerabilities reported in the CVE database while the final 4 rely both on reported and unreported vulnerabilities. For vulnerabilities not present in the CVE, it is not certain how the malware developer became aware of the vulnerability.

Malware	Year of Creation	Objective	No of CVE used	No of non-CVE vuln. used
Hydra	2008	D	0	0
Psybot	2009	D	0	1
Chuck Norris	2009	D	0	0
Tsunami	2010	D	0	0
Aidra	2012	D	0	1
Carna	2012	-	0	0
Bashlite	2014	D	0	0
Darlloz	2014	I	1	0
Spike	2014	D	0	0
TheMoon1	2014	-	0	0
Wifatch	2014	-	0	0
XOR	2014	D	0	0
Elknot	2015	D	0	0
Remaiten	2016	D	0	0
Hajime	2016	-	4	0
Mirai	2016	D	0	0
NewAidra	2016	D	0	0
LuaBot	2016	D	0	0
Amnesia	2017	D	1	0
BrickerBot	2017	S	0	0
IoTReaper	2017	D	9	4
Persirai	2017	D	2	0
Satori	2017	D	1	1
JenX	2018	D	3	0
TheMoon2	2018	P,S	4	0
VPNFilter	2018	S	14	3
Hide'n Seek	2018	S,I,M	10	3
Echobot	2019	D	73	19

Table 1. Recent IoT malware, their type and number of vulnerabilities exploited by each

The data we gathered is summarized in Tables 1 and 4.

Table 1 records every malware studied, and identifies for each how many vulnerabilities it exploits as part of its attack strategy, distinguishing vulnerabilities for which there exists a CVE record from those for which there are not. Table 1 also records the main objective of each IoT malware. These objectives are taken from a survey by Vignau et al. [58] and range over Denial of service attacks (D), Physical destruction of the target devise (P), income generation (i.e. cryptomining) (I), Spying (S) and Malware dissemination (M).

A number of observations are immediately obvious from an inspection of this data. As can be seen in Table 1, early IoT rarely exploited vulnerabilities, and never exploited multiple vulnerabilities. Starting in 2016, the use of vulnerabilities became more common, and some malware started exploiting multiple

vulnerabilities. This trend reached an apex with Echobot, a highly dynamic malware whose code is regularly updated with the inclusion of new exploits.

Vulnerabilities recorded in the NVD (those for which a CVE entry exists) outnumber unlisted vulnerabilities by a factor of 4, hinting that the NVD database is the preferred venue of malware developers to select exploitable vulnerabilities. However, in this respect, it is important to stress that some vulnerabilities only acquired a CVE entry after it appeared that IoT malware were exploiting these vulnerabilities [10]. A complete listing of every CVE exploited by each IoT that made use of recorded vulnerabilities is given in Table 3. The right-most column of the table identifies the references listing the vulnerabilities exploited by each malware.

It does not appear that the objective of the malware correlates with it's use of vulnerabilities as an infection vector, thought the limited size of the malware sample, as well as the fact that certain objectives are more common in later malware, makes a definitive determination difficult. In particular, spying and malware dissemination only occur in a single IoT malware each, both of them towards the end of the period of our study when the use of vulnerabilities had become more commonplace.

Finding 1:

IoT malware increasingly rely on exploiting vulnerabilities as part of their infection strategy. Recent malware is also much more likely to exploit multiple vulnerabilities. There is not enough evidence to conclude that malware with a specific objective is more likely to adopt this infection strategy. The NVD seems to be the preferred venue for malware developers to search for and find exploitable vulnerabilities.

4 Research Question 2

What types of CVEs are targeted by IoT malware developers?
In this section, we seek to determine if the CVEs targeted by IoT malware developers share distinctive characteristics that can help predict which CVEs are more likely to be targeted.

We found that 11 different malware made use of 99 CVEs a total 128 times (counting each use of a CVE by a different malware as distinct). Of these, 98 posses a CVSS V2 score and 64 possess a CVSS V3 score[3]. An analysis of this data indicates that the vulnerabilities targeted by IoT malware developers distinguish themselves in several ways.

Unsurprisingly, CVEs employed by IoT malware developers exclusively employ the 'Network' access vector, meaning that the vulnerable component is connected

[3] The only CVE for which no information was available, CVE-2013-5759, is a duplicate of CVE-2013-5758.

Table 2. CVEs exploited by each IoT botnet

Malware	Exploited CVEs	References
Darlloz	CVE-2012-1823	[3], [8], [48], [59]
Hajime	CVE-2016-10372, CVE-2018-10561, CVE-2018-10562, CVE-2015-4464, CVE-2018-7445, CVE-2018-14847, CVE-2013-6023	[65],[47], [22], [17] [25] [57]
Amnesia	CVE-2013-6023	[19] [63]
IoTReaper	CVE-2017-8225, CVE-2017-18377, CVE-2013-2678, CVE-2018-14933, CVE-2018-15716, CVE-2017-18378, CVE-2013-4980 CVE-2013-4981, CVE-2013-4982	[33] [52], [49],[64],[9], [44]
Persirai	CVE-2017-8225, CVE-2017-18377	[13],[54]
Satori	CVE-2014-8361	[39] [37] [36]
JenX	CVE-2017-18368, CVE-2017-17215, CVE-2014-8361	[45],[14]
TheMoon2	CVE-2018-1056, CVE-2018-14847, CVE-2018-10561, CVE-2018-10562	[26],[53], [2]
VPNFilter	CVE-2015-7261, CVE-2011-4723, CVE-2014-9583, CVE-2013-2678, CVE-2013-0229, CVE-2013-0230, CVE-2017-6361, CVE-2017-8877, CVE-2017-5521, CVE-2012-5958, CVE-2012-5959, CVE-2016-6277, CVE-2017-6549, CVE-2013-2679	[50] [27],[28],[15],[1]
Hide'n Seek	CVE-2016-10401, CVE-2017-8225, CVE-2017-18377, CVE-2018-14933, CVE-2018-15716, CVE-2017-18378, CVE-2013-4980,CVE-2013-4981, CVE-2013-4982, CVE-2013-2678	[7],[55],[4],[12],[5],[31]
Echobot	CVE-2003-0050, CVE-2005-0116, CVE-2005-2773, CVE-2005-2847, CVE-2005-2848, CVE-2006-2237, CVE-2006-4000, CVE-2007-3010, CVE-2008-3922, CVE-2009-0545, CVE-2009-2288, CVE-2009-2765, CVE-2009-5156, CVE-2009-5157, CVE-2010-5330, CVE-2011-3587, CVE-2011-5010, CVE-2012-0262, CVE-2012-4869, CVE-2013-3568,¸CVE-2013-4863, CVE-2013-5758, CVE-2013-5759, CVE-2013-5912, CVE-2013-5948, CVE-2013-7471, CVE-2014-3914, CVE-2014-8361,CVE-2015-2208, CVE-2015-4051, CVE-2016-0752, CVE-2016-10760, CVE-2016-1555, CVE-2016-6255, CVE-2016-6277, CVE-2017-14127, CVE-2017-14135, CVE-2017-16602, CVE-2017-16608, CVE-2017-18377, CVE-2017-5173, CVE-2017-5174, CVE-2017-6316, CVE-2017-6884, CVE-2017-8221, CVE-2017-8222, CVE-2017-8223, CVE-2017-8224, CVE-2017-8225, CVE-2018-1056, CVE-2018-10561, CVE-2018-10562, CVE-2018-11138, CVE-2018-11510, CVE-2018-14847, CVE-2018-14933, CVE-2018-15887, CVE-2018-17173, CVE-2018-20841, CVE-2018-6961, CVE-2018-7297, CVE-2018-7841, CVE-2019-12780, CVE-2019-12989, CVE-2019-12991, CVE-2019-14927, CVE-2019-14931, CVE-2019-15107, CVE-2019-16072, CVE-2019-17270 CVE-2019-18396 ,CVE-2019-2725, CVE-2019-3929.	[24],[42],[10],[23][40],[16],[11].

to the network stack. The alternative classifications are 'local', 'Adjacent Network' and 'Physical'.

The V2 scoring provides a 3-valued verdict as to the difficulty of exploiting each vulnerability. Performing a standard Khi-square test, we find with strong confidence ($p<.01$) that the selection of CVEs by malware creators is skewed towards easier CVEs. This result is even starker when considering the V3 scoring of this same element: only 3 out of 64(4.7%) CVEs for which a V3 score of complexity is provided are rated as 'HIGH' difficulty, versus 8.6% in the NVD database in general.

In addition to selecting attacks of lower complexity, IoT malware designers seem to prefer vulnerabilities with higher impact scores, as recorded in the V2 and V3 impact score metric. The average V2 and V3 impact scores for vulnerabilities exploited by IoT malware are 8.17 and 5.6 respectively, higher that the corresponding averages of 6.0 and 4.4 for the NVD database as a whole. Performing a standard Z-test [21] confirms that attackers indeed select CVE with high scores, for both V2 and V3.

The exploited CVEs also skew heavily towards CVEs that do not require user interaction : 93.8% of the exploited CVEs for which a V2 ranking of user interaction is provided do not require it, versus 68.6% for the NVD in general. Once again, a standard Z-test confirm the statistical significance of this result. The same result holds when considering the V3 score for user interaction: 92.1% of exploited malware did not require explicit user interaction versus 36.9% for the broader NVD.

Each CVE entry identifies a single CWE (Common Weakness Identification), that pinpoints the type of vulnerability in question. Once again, a Khi-square test indicates that the distribution of vulnerability types is not random, but seems to be skewed towards specific CWE types. The top six CWE types most frequently targeted by IoT malware developers are CWE-20 (Improper Input Validation), CWE-94 (Improper Control of Generation of Code), CWE-78 (Improper Neutralization of Special Elements used in an OS Command), CWE-77 (Improper Neutralization of Special Elements used in a Command), CWE-119 (Improper Restriction of Operations within the Bounds of a Memory Buffer) and CWE-287(Improper Authentication); together accounting for 67% of exploited vulnerabilities. Table 3 shows the top six most common CWE exploited by IoT malware, alongside with their frequency of occurrence among exploited CVEs and in the entire NVD database. Note that all six of these weaknesses are either input validation errors or authentication errors.

This result, however, should be qualified. Unlike the other information recorded in CVE entries, the proportion of CWE of each times varies widely from year to year, a factor we were unable to account for. Other limitations of the dataset are discussed in Section 6.

As can be seen in Table , some malware developers often look back several years in search of exploitable vulnerabilities. Notably, Echobot, discovered in June 2019, employed a vulnerability first uncovered in 2003. However, our analysis shows that most IoT malware tends towards the exploitation of recent vulnerabilities.

CWE	Proportion in IoT malware	Proportion in NVD
CWE-78	23.0%	8.6%
CWE-77	14.9%	2.5%
CWE-20	9.2%	1.1%
CWE-94	5.7%	0.5%
CWE-119	6.9%	12.5%
CWE-287	6.9%	2.1%

Table 3. Most frequently exploited CWE

In order to investigate further how the vulnerabilities exploited by IoT malware relate to the time of publication of the corresponding CVE entries, we computed the timespan that separates the public divulgation of a vulnerability and it's incorporation in IoT malware, for every CVE for which this information was available.

Unfortunately, only partial data was available in this regard. A malware may be updated multiple times, making it difficult to determine exactly when a given vulnerability was incorporated into its code. In fact, only for Echobot were we able to find multiple descriptions of the CVEs it exploits along with the exploitation date, allowing us to assign different CVEs to different versions of the malware. Furthermore, as discussed above, some vulnerabilities only received a CVE entry after it was found that a malware exploited this vulnerability. These and other limitations of the data are discussed in the Section 6. Nonetheless, the data that is available seems sufficient to draw broad conclusions.

Table 4 details the timespan, in months, that separates the publication of a CVE in the NVD database from its introduction in a malware. IoT malware for which this information could not be ascertained with sufficient confidence are omitted from the table. The information is grouped into the intervals of months given in the left most column. The center column gives the total number CVEs whose age was contained in each interval at the moment of their introduction in a malware while the right-most column breaks down this number by malware, using the following key : E: Echobot, D:Darlloz, A: Amnesia, V: VPNFilter.

In a December 2019 blogpost [41], Ruchna Nigam suggested that Echobot may be aiming at a 'sweetspot' of vulnerability exploitation by selecting both very recent vulnerabilities, for which the patch may not have been applied yet, as well as much older vulnerabilities, targeting systems that are not longer maintained. The data we gathered bears this analysis. Only 23% of the CVE for which we were able to obtain data were more than 6 months old, but less than 2 years old.

This infection strategy, however, seems unique to the concepts of Echobot, and every other IoT malware for which data was available seems to have picked vulnerabilities whose date of publication falls inside a fairly narrow range.

It is also noteworthy to see that multiple CVEs are exploited by several different IoT bots. Indeed, of the 98 CVEs in our corpus, 13 are exploited at least twice, 8 are exploited at least three times and 2 are exploited four times.

Time Interval (number of months)	Number of CVEs (total)	Number of CVEs by malware
≤ 5	12	E: 12
6-12	1	E:1
13-24	12	D :1 E :7 M : 4
25-72	19	A :1 S :1 E :10 V :7
73+	9	V :5 E :4

Table 4. Number of months that separate the publication of CVEs from their exploitation by IoT malaware.

This phenomenon is likely due to the large amount of code reused between IoT malware [58] and highlights the need for the prompt applications of security updates. Since the NVD database contains upwards of 6800 entries for software and firmware used by IoT devices, a CVE that has been exploited in the past is more than 8 times more likely to be exploited again in a different IoT malware.

Finding 2:

Malware developers are more likely to use CVE with high exploitability score, and low exploit difficulty. CWEs related to input validation errors or authentication errors are more likely to be targeted, with a small number of CWEs accounting for the plurality of exploited vulnerabilities. Malware developers tend to prefer recent CVEs, and are likely to reuse CVEs that have been targeted by other malware developers in the past.

5 Research Question 3

Drawing on the results of the previous section, we attempt to automatically ascertain which CVE are more likely to be targeted by IoT malware developers using a machine learning process. A successful classification will enable IoT developers and sysadmins to prioritize CVEs in the design and application of patches, proactively focusing on CVEs with a high likelihood of being incorporated into malware.

We performed a first filter over the NVD, eliminating any CVE whose associated products were not IoT devices. For this purpose, we compiled a list of any product that figures in any CVE exploited by any of the IoT malware in our dataset and excluded CVEs that did not include any product in the target list. At the end of this process, we had 6 300 entries.

For each of the remaining CVEs, we created a feature vector comprising the following datum of information: year of publication, CWE, Access Complexity (V2), Impact sub-score (V2), user interaction, exploitability Score (V2), and impact score. Each vector also indicated whether the CVE exclusively concerned products on the above mentioned list or additionally targeted products not listed.

The two classes are 'selected', for the CVEs that malware designer elected to target, and 'unsued', for other IoT vulnerabilities.

We used undersampling to overcome the unbalance in the dataset, aggregating the 98 CVEs used throughout this study with 200 randomly selected entries. We used 70% of the data for training and 30% for testing.

Algorithm	**Correctly Classified Instances**	**Incorrectly Classified Instances**	**Precision**	**Recall**
SVM	52 (81.25 %)	12 (18.75 %)	78.8%	83.9%
Random Forest	50 (78.1 %)	14 (21.9%)	79.3 %	74.2%
J48	53 (82.8 %)	11 (17.2 %)	83.3%	80.6%

Table 5. Classifier Results

The results are reported in Table 5.

These results indicate that the CVEs targeted by malware designers are sufficiently distinguished from those they avoid to allow automated detection with reasonable effectiveness. The J48 algorithm was particularly effective at predicting which CVEs will be selected while the other SVM was more effective at at ruling out CVEs unlikely to be targeted. It's clear that while the classification provides useful and actionable information, more research is needed before security researchers can confidently predict the future evolution of IoT malware.

Interestingly, we repeated the experiment with the elision of the 'year' datum in the feature vector and obtained similar results to those reported in Table 5. This indicates that the classification relies on core features of the CVEs, rather than on time spans that separates the publication of a CVEs from the creation of the malware studied.

Our use of undersampling to overcome the unbalance in the dataset is a threat to the validity of this result. This problem can be corrected as more data becomes available, a near certainty given the continued prevalence of IoT malware and the resourcefulness of IoT malware developers.

Finding 3:

The CVEs targeted by malware developers are sufficiently distinguished from those they avoid to allow automated detection with reasonable effectiveness, though more research is needed to refine the detection process.

6 Threats to Validity

We opted to include the entire corpus of vulnerabilities present in the NVD database in our analysis. A possible threat to validity derives from this decision,

since the NVD database contains vulnerabilities dating back from as early as 2002, and it is likely that most malware developers disregard such dated vulnerabilities. That said, some bots do include vulnerabilities several years old, such Echobot, deployed in 2019, which exploited a 16 year old vulnerability. We consequently opted to include the entire dataset.

The incompleteness of the data is another threat to validity. As discussed above, the V2 and V3 rating are not present for every vulnerability. Amongst the vulnerabilities utilized by IoT malware developers, a single one, CVE-2013-5759, utilized by Echobot, did not have either rating. Furthermore, a small number of CVEs may be duplicates, a fact we ignored. For example, CVE-2019-18396 is a duplicate report of the same vulnerability reported by CVE-2017-14127.

The fact that the average impact score, difficulty of exploitation and the distribution of CWEs in the NVD varies from year to year is a threat to the validity of the results presents in Section 4 since the vulnerabilities exploited by IoT designers skew towards the more recent past. The differences we observed in the CVSS scores and CWEs of exploited CVEs may be caused in part by drift in the the values values over the years.

A treat to the validity of our results exists because some vulnerabilities received a CVE entry only after they began to be exploited by an IoT malware, especially Echobot (see for e.g. [10]). We were unable to identify these vulnerabilities with certainty, and this fact may have led us to understate the number of vulnerabilities without a corresponding NVD entry that are exploited by IoT malware (Section 3) and to overstate the propensity of malware designer to target recent vulnerabilities (Section 4).

As mentioned above, the unbalance present in the dataset, and our use of undersampling to overcome it, is a threat to the validity of the results reported in Section 5.

7 Related Works

Recently, Blinowski et al. [6] proposed a classification of vulnerabilities associated with IoT devices, extracted from the NVD database. They manually grouped vulnerability records into seven categories (Home, mobile devices, etc.). This classification was used later to train an SVM to predict the category of new vulnerabilities. Their approach achieves a precision and recall of 70%-80% for categories for which they have a large number of vulnerabilities, and of 50% accuracy or less for less-populated categories.

Li et al. [30] proposed a vulnerability mining algorithm to analyze and obtain essential characteristics of software vulnerability-based data mining techniques. Their algorithm was used on software vulnerabilities using the NVD dataset. When applied to detecting vulnerabilities in three projects, their approach achieved a recall of around 70% and precision of 60%.

Spanos and Angelis [51] presented a model that can automatically predict the characteristics of vulnerabilities. This is an important task since the specification of vulnerabilities is used to determine their severity, complexity, impact, and other

characteristics, used by vulnerability scoring systems. Their model combines text processing and multi-target classification technique. They applied their model to a dataset of 99,000 vulnerability records from the NVD. The results vary depending on the vulnerability characteristic that is the subject of prediction and the algorithm used for classification, with an F-measure ranging between 42.88% and 67.91%.

Le et al. [29] proposed an approach to automatically assess software vulnerabilities with concept drift using software vulnerability descriptions. Their approach combines both character and word features. They applied their approach to the prediction of seven vulnerability characteristics. They experimented with more than 100,000 vulnerabilities from NVD and showed that their approach can predict vulnerability effectively without having to retrain the models, which suggest that their models can be used to overcome the problem of concept drift.

Wijayasekara et al. [61] focused on so-called hidden impact vulnerabilities, i.e. vulnerabilities that appeared long after the associated bugs have been made public. They develop a text mining classifier to identify hidden impact vulnerabilities from bug report databases. The authors extended this work in[60,62] by using information gain and genetic algorithms [60] and three different classifiers (NaïveBayes, Naive Bayes Multinomial, and C4.5 Decision Tree) [62].

Murtaza et al. [34] conducted an empirical study to understand the trends of software vulnerabilities over time, the common patterns of software vulnerabilities, and whether or not one can predict the type of vulnerability in a software application. They used NVD and their main source of data to mine six years of software vulnerabilities from 2009 to 2014. They found that the patterns of vulnerability events follow the first order Markov property, i.e., the next vulnerability can be predicted by the previous vulnerability. They also found that the next vulnerability can be predicted with approximately 90% precision and 80% recall, just by using the previous vulnerability. finally, they found that collectively mobile applications have higher vulnerabilities than traditional software applications.

Na et al. [35] proposed a classification method for categorizing CVE entries into vulnerability type using naïve Bayes classifiers. They showed that their approach can analyze CVE entries that are not yet classified. Frei et al. [20] conducted a study in which they examine the time of discovery of vulnerabilities, the time of disclosure of attacks, and the time of availability of patches. Their study uses mainly NVD data. They found that software vendors are slow to provide patches despite the fact that attacks that exploit zero-day vulnerabilities are an increasing trend. Neuhaus and Zimmermann [38] proposed an approach to automatically categorize CVE vulnerabilities into vulnerability types by using Latent Dirichlet Allocation (LDA).

Valente et al. [56] analyze various types of IoT devices and uncover the vulnerabilities they contain. They found 9 new CVEs that can be employed to perform new kinds of attacks including drone hijacking, remote sexual assault, or harassment. They classify the CVEs exploited by IoT malware in 4 categories depending on the interactions between the attackers and the IoT devices.

8 Conclusions

In this study we investigated the use of CVEs by IoT malware developers. We found that IoT malware increasingly relies upon the exploitation of vulnerabilities as an infection vector, and that the NVD seems to be the preferred source to obtain these vulnerabilities. We also found that the vulnerabilities selected by IoT malware differ from the broader NVD in several respects. Notably, IoT malware developers then to prefer vulnerabilities with lower than average exploitation complexity and higher than average impact. Targeted vulnerabilities are remotely exploitable and less likely to necessitate user interaction. Certain specific CWEs, reflecting input validation and authentication errors, are also more likely to be targeted. Indeed, the CVEs targeted by malware designer are sufficiently distinct to permit automated prediction using machine learning algorithms.

In addition, we compiled data about the use of vulnerabilities by IoT malware, which may be useful to security researchers in the future.

References

1. Vpnfilter-affected devices still riddled with 19 vulnerabilities.
2. Netlab 360. Gpon exploit in the wild (iv) - themoon botnet join in with a 0day(?). https://blog.netlab.360.com/gpon-exploit-in-the-wild-iv-themoon-botnet-join-in-with-a-0day/, 2018.
3. Kishore Angrishi. Turning internet of things(iot) into internet of vulnerabilities (iov) : Iot botnets. 02 2017.
4. Liviu Arsene. Hide and seek iot botnet learns new tricks: Uses adb over internet to exploit thousands of android devices. https://labs.bitdefender.com/2018/09/hide-and-seek-iot-botnet-learns-new-tricks-uses-adb-over-internet-to-exploit-thousands-of-android-devices/, September 2018.
5. Avast. Let's play hide 'n seek with a botnet. https://blog.avast.com/hide-n-seek-botnet-continues, December 2018.
6. Grzegorz Blinowski and Pawel Piotrowski. Cve based classification of vulnerable iot systems. In *Theory and Applications of Dependable Computer Systems (DepCoS-RELCOMEX 2020)*, pages 82–93|, 2020.
7. Bogdan BOTEZATU. New hide 'n seek iot botnet using custom-built peer-to-peer communication spotted in the wild. https://labs.bitdefender.com/2018/01/new-hide-n-seek-iot-botnet-using-custom-built-peer-to-peer-communication-spotted-in-the-wild/, January 2018.
8. Biagio Botticelli. IoT honeypots: State of the art. https://fr.slideshare.net/BiagioBotticelli/state-of-the-art-iot-honeypots, 2017.
9. Krebs Brian. Fear the reaper, or reaper madness ? https://krebsonsecurity.com/2017/10/fear-the-reaper-or-reaper-madness/more-41321, 2017.
10. Larry Cashdollar. Latest echobot: 26 infection vectors. blogs.akamai.com/sitr/2019/06/latest-echobot-26-infection-vectors.html, 2019.
11. Catalin Cimpanu. Le nouveau malware d'echobot est un concentré de vulnérabilités. https://www.zdnet.fr/actualites/le-nouveau-malware-d-echobot-est-un-concentre-de-vulnerabilites-39886143.htm, 2019.

12. Adrian Şendroiu and Vladimir Diaconescu. Hide'n'seek: an adaptive peer-to-peer iot botnet. Virus Bulletin, October 2018.
13. New Jersey Cybersecurity and Communications Integration Cell. Persirai. https://www.cyber.nj.gov/threat-profiles/botnet-variants/persirai, 2017.
14. Asher Davila. Jenx botnet: A new iot botnet threatening all. https://unit42.paloaltonetworks.com/home-small-office-wireless-routers-exploited-to-attack-gaming-servers/, 2019.
15. Talos Unit Edmund Brumaghin. VPNFilter iii: More tools for the Swiss army knife of malware. https://blog.talosintelligence.com/2018/09/vpnfilter-part-3.html, 2018.
16. Kreminchuker Eli, Zavodchik Maxim, and Pompon Raymond. Echobot malware now up to 71 exploits, targeting scada. https://www.f5.com/labs/articles/threat-intelligence/echobot-malware-now-up-to-71-exploits–targeting-scada, 2019.
17. Radware Emergency Response Team. Hajime - friend or foe ? https://security.radware.com/ddos-threats-attacks/hajime-iot-botnet/, 2017.
18. FIRST. Common vulnerability scoring system version 3.1, 6 2019.
19. Charles Frank, Cory Nance, Sam Jarocki, Wayne E Pauli, and SD Madison. Protecting iot from mirai botnets; iot device hardening. In *Proceedings of the Conference on Information Systems Applied Research, Austin, TX, USA*, page 1508, 2017.
20. Stefan Frei, Martin May, Ulrich Fiedler, and Bernhard Plattner. Large-scale vulnerability analysis. In *Proceedings of the 2006 SIGCOMM Workshop on Large-Scale Attack Defense*, LSAD '06, page 131–138, New York, NY, USA, 2006. Association for Computing Machinery.
21. J.E. Freund, I. Miller, and M. Miller. *John E. Freund's Mathematical Statistics with Applications*. Pearson Education. Prentice Hall, 2004.
22. Stephen Herwig, Katura Harvey, George Hughey, Richard Roberts, and Dave Levin. Measurement and analysis of hajime, a peer-to-peer iot botnet. In *NDSS*, 2019.
23. Ionut Ilascu. Echobot botnet spreads via 26 exploits, targets oracle, vmware apps. https://www.bleepingcomputer.com/news/security/echobot-botnet-spreads-via-26-exploits-targets-oracle-vmware-apps/, 2019.
24. Ionut Ilascu. New echobot botnet variant uses over 50 exploits to propagate. https://www.bleepingcomputer.com/news/security/new-echobot-botnet-variant-uses-over-50-exploits-to-propagate/, 2019.
25. Van Der Wiel Jornt, Diaz Vicente, Namestnikov Yury, and Konstantin Zykov. Hajime, the mysterious evolving botnet. https://securelist.com/hajime-the-mysterious-evolving-botnet/78160/, 2017.
26. BlackLotus Lab. A new phase of themoon. https://blog.centurylink.com/a-new-phase-of-themoon/, 2019.
27. William Largent. New VPNFilter malware targets at least 500k networking devices worldwide. blog.talosintelligence.com/2018/05/VPNFilter.html, 2018.
28. William Largent. VPNFilter update - VPNFilter exploits endpoints, targets new devices. blog.talosintelligence.com/2018/06/vpnfilter-update.html, 2018.
29. Triet Huynh Minh Le, Bushra Sabir, and M. Ali Babar. Automated software vulnerability assessment with concept drift. In *Proceedings of the 16th Inter. Conf. on Mining Software Repositories*, MSR '19, page 371–382. IEEE Press, 2019.
30. X. Li, J. Chen, Z. Lin, L. Zhang, Z. Wang, M. Zhou, and W. Xie. A mining approach to obtain the software vulnerability characteristics. In *2017 Fifth International Conference on Advanced Cloud and Big Data (CBD)*, pages 296–301, 2017.
31. MalwareTech. Tracking the hide and seek botnet. https://www.malwaretech.com/2019/01/tracking-the-hide-and-seek-botnet.html, January 2019.

32. Peter Mell, Karen Scarfone, and Sasha Romanosky. A complete guide to the common vulnerability scoring system. Technical report, National Institute of Standards and Technology and Carnegie Mellon University, 2007.
33. Priscilla Moriuchi and Sanil Chohan. Mirai-variant iot botnet used to target financial sector in january 2018. https://go.recordedfuture.com/hubfs/reports/cta-2018-0405.pdf, 2018.
34. Syed Shariyar Murtaza, Wael Khreich, Abdelwahab Hamou-Lhadj, and Ayse Basar Bener. Mining trends and patterns of software vulnerabilities. *J. Syst. Softw.*, 117(C):218–228, July 2016.
35. Sarang Na, Taeeun Kim, and Hwankuk Kim. A study on the classification of common vulnerabilities and exposures using naïve bayes. In *BWCCA*, 2016.
36. Netlab360. Botnets never die, satori refuses to fade away. blog.netlab.360.com/botnets-never-die-satori-refuses-to-fade-away-en/, 2018.
37. Netlab360. Warning: Satori, a mirai branch is spreading in worm style on port 37215 and 52869. blog.netlab.360.com/warning-satori-a-new-mirai-variant-is-spreading-in-worm-style-on-port-37215-and-52869-en/, 2018.
38. S. Neuhaus and T. Zimmermann. Security trend analysis with cve topic models. In *2010 IEEE 21st International Symposium on Software Reliability Engineering*, pages 111–120, 2010.
39. NewSkySecurity. Masuta : Satori creators' second botnet weaponizes a new router exploit. https://blog.newskysecurity.com/masuta-satori-creators-second-botnet-weaponizes-a-new-router-exploit-2ddc51cc52a7, 2018.
40. Ruchna Nigam. Mirai variant echobot resurfaces with 13 previously unexploited vulnerabilities. https://unit42.paloaltonetworks.com/mirai-variant-echobot-resurfaces-with-13-previously-unexploited-vulnerabilities/, 2019.
41. Ruchna Nigam. Mirai variant echobot resurfaces with 13 previously unexploited vulnerabilities, 12 2019.
42. Ruchna Nigam. New mirai variant adds 8 new exploits, targets additional iot devices. https://unit42.paloaltonetworks.com/new-mirai-variant-adds-8-new-exploits-targets-additional-iot-devices/, 2019.
43. NIST. National vulnerability database - general information, 8 2020.
44. Check Point. Iotroop botnet: The full investigation. https://research.checkpoint.com/iotroop-botnet-full-investigation/, 2017.
45. Radware. Jenx botnet: A new iot botnet threatening all. https://security.radware.com/ddos-threats-attacks/threat-advisories-attack-reports/jenx/, 2018.
46. Khaled Salah and Minhaj Khan. Iot security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 11 2017.
47. Edwards Sam and Profetis Ioannis. Hajime: Analysis of a decentralized internet worm for iot devices. www.cs.umd.edu/class/fall2017/cmsc818O/papers/hajime-rapidity.pdf, 2016.
48. James Scott Sr and Winter Summit. Rise of the machines: The dyn attack was just a practice run december 2016. 2016.
49. FortiGuard SE. Reaper: The next evolution of IoT botnets. https://www.fortinet.com/blog/threat-research/reaper-the-next-evolution-of-iot-botnets.html, November 16, 2017.
50. Sapalo Sicato, Jose Costa, Pradip Kumar Sharma, Vincenzo Loia, and Jong Hyuk Park. Vpnfilter malware analysis on cyber threat in smart home network. *Applied Sciences*, 9(13):2763, 2019.
51. G. Spanos and L. Angelis. A multi-target approach to estimate software vulnerability characteristics and severity scores. *J. Syst. Softw.*, 146:152–166, 2018.

52. Trend Mirco System. New rapidly-growing iot botnet - reaper. https://success.trendmicro.com/solution/1118928-new-rapidly-growing-iot-botnet-reapercollapseTwo, 2018.
53. Seals Tara. Themoon rises again, with a botnet-as-a-service threat. https://threatpost.com/themoon-botnet-as-a-service/141393/, 2019.
54. TrendMicro. Persirai: New internet of things (iot) botnet targets ip cameras. https://blog.trendmicro.com/trendlabs-security-intelligence/persirai-new-internet-things-iot-botnet-targets-ip-cameras/, 2017.
55. TrendMicro. "hide 'n seek" botnet uses peer-to-peer infrastructure to compromise iot devices. https://www.trendmicro.com/vinfo/es/security/news/internet-of-things/-hide-n-seek-botnet-uses-peer-to-peer-infrastructure-to-compromise-iot-devices, January 2018.
56. Junia Valente, Matthew A Wynn, and Alvaro A Cardenas. Stealing, spying, and abusing: Consequences of attacks on internet of things devices. *IEEE Security & Privacy*, 17(5):10–21, 2019.
57. Julio Velasquez. Hajime botnet variant, 12 2018.
58. B. Vignau, R. Khoury, and S. Hallé. 10 years of iot malware: A feature-based taxonomy. In *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 458–465, July 2019.
59. Aohui Wang, Ruigang Liang, Xiaokang Liu, Yingjun Zhang, Kai Chen, and Jin Li. An inside look at iot malware. In *International Conference on Industrial IoT Technologies and Applications*, pages 176–186. Springer, 2017.
60. D. Wijayasekara, M. Manic, and M. McQueen. Information gain based dimensionality selection for classifying text documents. In *2013 IEEE Congress on Evolutionary Computation*, pages 440–445, 2013.
61. D. Wijayasekara, M. Manic, J. L. Wright, and M. McQueen. Mining bug databases for unidentified software vulnerabilities. In *2012 5th International Conference on Human System Interactions*, pages 89–96, 2012.
62. Dumidu Wijayasekara, Milos Manic, and Miles McQueen. Vulnerability identification and classification via text mining bug databases. In *IECON 2014 - 40th Annual Conf. of the IEEE Industrial Elec. Society, Dallas, TX, USA*, pages 3612–3618.
63. Claud Xiao and Unit 42 Cong Zheng. New IoT/Linux malware targets dvrs, forms botnet. https://unit42.paloaltonetworks.com/unit42-new-iotlinux-malware-targets-dvrs-forms-botnet/, 2017.
64. yegenshen. IoT_reaper: A rappid spreading new iot botnet, 10 2017.
65. Andrei Costin Zaddach and Jonas. IoT malware comprehensive survey, analysis framework and case studies. In *Black Hat 2018*.